**A vanádium-dioxid modellezési kérdései**

**A hőkapacitás modellje**

A hőkapacitás definíciója:

(1)

Azaz az anyagban tárolt energia (*Eb*) megváltozása a hőmérséklet-változás hatására.

A VO2 fázisátalakulásához 236,5⋅106 J/m3 energia szükséges: növekvő hőmérsékletnél elnyelődik, csökkenő hőmérsékletnél felszabadul. Magától értetődő lenne ezt egy hőárammal modellezni, mint időegység alatti energiaváltozást, azonban ez a megoldás szinte biztosan numerikus instabilitáshoz vezetne. (A fázisváltáshoz szükséges energia sokszorosa a hőkapacitásban tárolt energia megváltozásához képest, így a számítás során hatalmas hőmérséklet-ugrások jelennének meg.) A stabil számítás érdekében ezért, a hőkapacitáshoz hasonlóan, implicit Euler módszert alkalmazunk.

Feltételezzük, hogy egy lépésben a VO2 hőmérséklete vagy csak nő, vagy csak csökken, így egy lépésen belül a hiszterézissel nem kell külön foglalkozni, minden lépés kiinduló hőmérsékletéhez egy egyértékű függvény adható, amely a hőmérséklet növekedése vagy csökkenése esetén is érvényes. A következő lépéshez új függvényt adunk.

Az anyag a fázisátalakulás miatt az *F(T)* energiát veszi fel:

(2)

Ahol *K(T)* egy olyan monoton függvény, amely *T1*-ben 0, *T2*-ben 1. (*EA* a fázisváltáshoz szükséges energia.) Hiszterézis esetén a növekvő és a csökkenő hőmérséklethez tartozó *T1* és *T2* különbözik egymástól, és *K(T)* akár jellegében is különbözhet, nem csak *T1*-ben és *T2*-ben. Ezt nem jelöljük külön.

Az anyagban tárolt energia:

(3)

Az energia megváltozása *TA* és *TB* hőmérséklet között:

(4)

Definiáljuk a bővített hőkapacitást (1)-hez hasonlóan, de (3) alapján:

(5)

Vezessünk be két egyszerűsítést:

1. A VO2 hőmérsékletfüggése nem ismert. Azoknál a szilárd anyagoknál, amelyek fajhőjének ismert a hőmérsékletfüggése, az jellemzően 1-4 ezrelék °C-onként szobahőmérsékleten, ezért a VO2 hőkapacitását konstansnak tekintjük. Ismert hőfokfüggés esetén is tekinthetjük egy szimulációs lépésben konstansnak a hőkapacitást, majd a következő lépésben az annak kezdetére kiszámított hőmérsékleten vesszük azt figyelembe.
2. A *K(T)* függvény sem ismert VO2 esetében. Az egyszerűség érdekében tekintsük lineárisnak. (Ha ismertté válik a tényleges függvény, a számításban probléma nélkül lecserélhetjük arra. A gyakorlatban az eddigi tapasztalatok alapján, ha a VO2 egy része eléri a fázisváltáshoz szükséges hőmérsékletet, rövid idő alatt kialakul a csatorna, melynek hőmérséklete lényegesen a fázisváltás sávja fölé kerül, így valószínűleg a görbe alakjának nincs jelentősége.) Az alkalmazott *K(T)* függvény tehát a következő:

(6)

Ezt behelyettesítve a bővített hőkapacitás:

(7)

Az anyagban tárolt energia az egyszerűsítések behelyettesítésével:

(8)

A cellában tárolt energia időegység alatti megváltozása, azaz a hőáram:

(9)

Ha a hőkapacitást két időpont között konstansnak tekintjük, ami az időpontokban ugrik, akkor is ugyanez az eredmény.

*CB*-t (7)-ből behelyettesítve:

(10)

Az implicit Euler integrálás:

(11)

(12)

(13)

Az (n+1)-et elhagyjuk a jelölésből, így a *P* és *T* a kapacitás aktuális lépésbeli hőáramát és hőmérsékletét jeleníti meg:

(14)

*CB/Δt* egy hővezetés, *gC*, a második tag egy hőáram, *PC* = *gCT(n)*:

(15)

*P*

*T*

*gC*

*PC*

Szemléletesebb azonban, ha Norton-Thevenin átalakítással feszültségforrást teszünk be:

*P*

*T*

*gC*

*T(n)*

Azaz a kapacitás az előző lépés hőmérsékletéhez adja hozzá a *gC*-n eső hőmérsékletet (negatív is lehet).

A VSUN3-ba épített megoldás ezt a modellt használja. A modell problémája, hogy a fázisátalakulás hőmérséklet-tartományában a bővített hőkapacitás sokszorosára nő. Pl. egy modellben a *Cth* = 1,81⋅106 J/m3K ha *T* < *T1*, 3,54⋅106 J/m3K, ha *T* > *T2*, viszont *T1* és *T2* között, mivel a modellben *T2-T1* = 9°C, 236,5⋅106 / 9 = 26,3⋅106 J/m3K adódik *Cth*-hoz, azaz egy nagyságrend az ugrás (1. ábra). Ezt a nagy hőkapacitást mindenképpen figyelembe kell venni. Mivel a fázisátalakulás során az elektromos vezetés gyorsan csökken, általában egy szimulációs lépés alatt kialakul a csatorna, melynek hőmérséklete jóval a fázisátalakulás sávja fölött adódik. Így a diszkrét időpontokban való mintavételezés miatt a nagyobb hőkapacitású tartományt a szimuláció során teljesen figyelmen kívül hagynánk. A VSUN3 úgy oldja meg a problémát, hogy visszalép időben, és addig csökkenti az időlépést, hogy a fázisváltó tartomány elejére kerüljünk, és a fázisváltó tartományból kilépni is csak úgy enged, ha a fázisváltó tartomány vége közelében vagyunk (10%-on belül).

*1. ábra: CB hőmérsékletfüggése VO2 esetén*

Ennek a megközelítésnek több problémája van:

* Vissza kell lépni időben, és kisebb időlépéssel megismételni a számítást. Ez jó pár többlet redukciós lépést igényel, azaz a számítás időigénye nagy lesz.
* A 10%-os tartomány nagy numerikus hibát okoz.
* Minden egyes elemi cella fázisváltásakor végre kell hajtani a lépésköz csökkentést, ami összességében nagyon sok plusz lépést eredményez.

A VSUN3 algoritmusának további problémája, hogy szukcesszív approximációt használ a nemlineáris egyenletek megoldásához, és annak érdekében, hogy ne kezdjen oszcillálni, a fázisváltás tartományában nagyon le kell csökkenteni a lépésközt. (Ez nem a hőkapacitás, hanem az elektromos vezetés gyors változása miatt van.) Ezt a gondot remélhetőleg megoldja majd a Newton-Raphson algoritmus.

A V6-hoz a hőkapacitás számításának a következő módosítását próbáljuk meg: az anyag által elnyelt/felszabadított energiát ne *CB*-nek a szimuláció pillanatában érvényes értéke alapján számoljuk, hanem a két időpont közötti változásokat is vegyük figyelembe.

Az anyagban tárolt energia megváltozását (4) írja le. *TB* természetesen nem ismert a szimulációs lépés kezdetekor, ezért az első iterációban *TA*-val számolunk, de az iteráció végére a *TB* hőmérséklet fog kiadódni, ha konvergens a számítás. Ha nem konvergens, jöhet az időlépés csökkentése, ez azonban remélhetőleg sokkal ritkábban lesz szükséges, mint a VSUN3 esetében.

A (4) egyenlet alkalmazását a gyakorlatban a *CB* bővített hőkapacitás módosított meghatározásával végezzük, azaz (5) helyett (16)-ot alkalmazzuk:

(16)

Itt *ΔF*-hez a *K* függvény (6) egyszerű alakját alkalmazzuk, a *Cth*-nál nem konstanssal számolunk, hanem a kezdő és véghőmérsékleten érvényes értékének átlagát mint integrálközelítést alkalmazzuk. A korábban bemutatott VO2 modell esetén *CBm* hőmérsékletfüggése a 2. ábrán látható.

*2. ábra: CBm hőmérsékletfüggése VO2 esetén, a következő lépés három különböző kezdőhőmérséklete esetén.*

Ami fontos különbség *CB*-hez képest, hogy *CBm* meghatározásához két különböző hőmérséklet szükséges, *CB* meghatározásához csak egy. Ennek megfelelően az 1. ábra *x* tengelye mutatja ezt az egy hőmérsékletet, míg a 2. ábra *x* tengelye a lépés végének hőmérsékletét adja, a lépés kezdőhőmérsékletét három konkrét értékre mutatja a grafikon:

* Ha *TA*=50°C-on indulunk, akkor a fázisváltás 61°C-os kezdőhőmérséklete alatt az ott érvényes 1,81e6-os hőkapacitást kapjuk. 61 és 70°C között folyamatosan nő a bővített hőkapacitás a fázisváltás által elnyelt hő (és részben a hőkapacitás növekedése) miatt. 70°C fölött csökkenni kezd a bővített hőkapacitás, mert itt a hőkapacitás 3,54e6-os értékéhez tart az átlag.
* A fázisváltás kb. közepén, *TA*=65°C-on indulunk. A kiindulópontban a *CBm* megegyezik a *CB*-vel. A fázisváltás tartományának többi pontjában az egyezés nem pontos, mert nem az adott hőmérséklethez tartozó *Cth*, hanem a kiinduló és a végponton érvényes *Cth* átlaga adódik *F*-hez.
* *TA*=75°C-on indulunk. Lényegében az első eset inverzéről van szó.

Érdekességképpen: a 3. ábra szélesebb hőmérséklet-tartományon mutatja be ugyanezt. A szimuláció során ekkora hőmérséklet-ugrások legfeljebb az iterációban, köztes értékként fordulhatnak elő, mert a pontosabb számítás érdekében az egy lépésben megengedett maximális hőmérséklet-változás korlátozva lesz oly módon, hogy ha egy küszöböt meghalad a hőmérséklet-változás, akkor a szimulációs lépésközt csökkenteni fogjuk. A küszöböt majd a gyakorlati tapasztalatok alapján állítjuk be.

*3. ábra: a 2. ábra nagyobb x tartományon.*

A bemutatott példában hiszterézismentes esetet vizsgáltunk.

**Hiszterézis számítása**

*4. ábra: Hiszterézisparaméterek magyarázata.*

A *K(T)* függvény helyett egy olyan *H(T,HA)* függvényt használunk, amelyik a fázisváltás mértékét adja az adott cellára 0 és 1 között. A kiinduló hőmérséklet és fázismérték: *TA* és *HA*, az új: *TB* és *HB*. A hiszterézis jellemzői: *TH1*, *TH2*, *SZ*, pl. 58, 66, 3. Származtatott hőmérsékletek: *TK*=*TH1-SZ*, *TV*=*TH2+SZ*, *TN*=*TH1+SZ*, *TM*=*TH2-TH1*, pl. 55, 69, 61, 8. A függvény:

*HB* = *H*(*TB*, *HA*){

Ha *TB* <= *TK*, akkor

Return 0

Ha *TB* >= *TV*, akkor

Return 1

*TC* = *TH1* + *HA*\**TM*

Ha *TB* < *TC* – *SZ*, akkor

Return (*TB* – *TK*) / *TM*

Ha *TB* <= *TC* + *SZ*, akkor

Return *HA*

Return (*TB* – *TN*) / *TM*

}

A hőkapacitás:

*CBm* = CBm(*TA*, *TB*, *HA*, *HB*){

Ha *TA* == *TB*, akkor

Return *HA* == 0 || *HA* == 1 ? *CthA* : *CthA* + *EA* / *TM* // azaz *CB*

Return *EA* \* (*HB* – *HA*) / (*TB* – *TA*) + (*CthA* + *CthB*) / 2

}

A kiindulópontban, amikor véghőmérsékletnek a kiinduló hőmérsékletet adjuk meg, *CB* lesz *CBm*. Ha a kiindulópont egy részben fázisátalakult cellát jelez, akkor *CB* tartalmazza a fázisátalakulásból eredő hőkapacitás részt is. Akkor is, ha utána a fázisátalakulás nem folytatódik, azaz csak az alap hőkapacitást kellene figyelembe venni. Ennek oka, hogy

* a számítás kezdetén nem tudjuk, hogy folytatódik-e a fázisátalakulás, ez csak az iterációk során derül ki.
* nagy valószínűséggel folytatódik a fázisátalakulás, azaz nem fordul meg a hőmérséklet-változás iránya az előző lépéshez képest.
* ha mégsem folytatódik a fázisátalakulás, akkor sem baj, hogy az első iterációban nagyobb bővített hőkapacitással számolunk, mert az csillapítja a hőmérséklet-változás mértékét, és a következő iterációban már a jó bővített hőkapacitást vesszük majd figyelembe.

A *CBm* függvény nem hiszterézises fázisátalakuláskor is használható, ha *HA* és *HB* helyére *K(TA)*-t ill. *K(TB)*-t tesszük be. Sőt, a *H* függvény is használható, ha *SZ*=0-t állítunk be.

A *CBm* viselkedését az 5. ábrán figyelhetjük meg hiszterézises esetben, a fejezet elején bemutatott hiszterézisparaméterek esetén. A hiszterézismentes és hiszterézises viselkedés különbségét legjobban a 2. és 5. ábra narancssárga görbéinek összehasonlításával láthatjuk. Mindkét esetben a felfutó szakasz közepén indulunk, félig fázisváltott anyaggal. Hiszterézis nélkül bármely irányban változik a hőmérséklet, a fázisváltás is folytatódik valamelyik irányban, emiatt *CBm* mindkét irányban magas marad. Hiszterézises esetben csak a növekvő hőmérséklet irányában folytatódik a fázisváltás, csökkenő hőmérséklet esetében 6°C-ot kell lejjebb menni, hogy a fázisváltás újrainduljon, addig csak *Cth* van jelen.

A citromsárga, szaggatott görbe azt az esetet mutatja be, amikor nem a felfutó vagy a lefutó görbén indulunk, hanem a kettő között, jelen esetben a 4. ábra piros, szaggatott vonallal jelzett szakaszának közepén. Ebben az esetben bármerre lépünk, nincs fázisváltás, csak 3°C után, így itt a *CBm*-et a *Cth* adja. A kiindulópontnál egy tüskét látunk, amit az okoz, hogy *TA*==*TB* esetben *CBm* = *CB*. Vizsgálhatnánk, hogy a kiindulópont rajta van-e a felfutó vagy lefutó élen, és ha nincs, akkor itt csak *Cth*-t adjon, de nem érdemes erre CPU időt áldozni.

*5. ábra: CBm hiszterézis esetén.*

Az értékek kiszámításához használt program:

#include <stdio.h>

double TH1, TH2, SZ, TK, TV, TN, TM, ITM, EA;

ertek\_t H(double TB, double HA) {

if (TB <= TK)

return ertek\_t(0, 0);

if (TB >= TV)

return ertek\_t(1, 0);

double TC = TH1 + HA\*TM;

if (TB < TC - SZ)

return ertek\_t((TB - TK) \* ITM, ITM);

if (TB <= TC + SZ)

return ertek\_t(HA, 0);

return ertek\_t((TB - TN) \* ITM, ITM);

}

double CBm(double TA, double TB, double HA, double HB, double CthA, double CthB) {

if (TA == TB)

return HA == 0 || HA == 1 ? CthA : CthA + EA / TM;

return EA\*(HB - HA) / (TB - TA) + (CthA + CthB) / 2;

}

void strreplace(char \*s, char mit, char mire) {

for (; \*s != '\0'; s++)

if (\*s == mit)

\*s = mire;

}

int main() {

char puff[1024];

EA = 236.5;

TH1 = 58;

TH2 = 66;

SZ = 3;

TK = TH1 - SZ;

TV = TH2 + SZ;

TN = TH1 + SZ;

TM = TH2 - TH1;

ITM = 1.0 / TM;

double TA = 62;

double HA = 0.5;

double CthA = 1.81 + (3.54 - 1.81)\*HA;

for (double TB = 40; TB < 90.1; TB++) {

double HB = H(TB, HA);

double CthB = 1.81 + (3.54 - 1.81)\*HB;

double cbm = CBm(TA, TB, HA, HB, CthA, CthB);

sprintf(puff, "%g %g %g\n", HB, CthB, cbm);

strreplace(puff, '.', ',');

printf("%s", puff);

}

return 0;

}

**Elektromos vezetés**

A VSUN3 a 4. ábrához hasonló módon, paralelogramma alakú hiszterézises tartomány szimulációját teszi lehetővé. A V6-ban a H függvénnyel különválik a fázisváltás mértékének meghatározása. Ezzel nagyon egyszerű a paralelogramma alakú hiszterézis megvalósítása, mert csak H arányában kell a hiszterézis előtti és utáni vezetés lineáris kombinációját kiszámítani. Megoldható azonban az is, hogy ne paralelogramma legyen a hiszterézis, hanem pl. ilyen:

*6. ábra: Elektromos vezetés lehetséges hiszterézise.*

Ebben az esetben a felső és alsó részre külön törtvonalas függvényt definiálunk, amelyeknek a hiszterézis hőmérsékleti tartományára átlapolódó módon definiálunk. Vagyis a 6. ábrán a kisvezetésű értékeket 69°C-ig, a nagyvezetésű értékeket 55°C-ig megadjuk, és az adott hőmérsékleten érvényes értékük lineáris kombinációját számoljuk H arányában. Megadás a v6\_core bemenő fájljában:

PGENB3T25;200;L700;L2e5;

Az L jelzi, hogy a megadott érték a hiszterézis tartomány másik szélén érvényes, azaz L700; 69 foknál, L2e5; pedig 55 foknál.

Az anyagparaméterek vissza kell adják a hőmérséklet szerinti deriváltjukat is a Newton-Raphson algoritmushoz. Ehhez szükség van a hiszterézisgörbe meredekségére is: *m* = 1/*TM*. Ha VSUN3-as hiszterézismodellt használnánk, akkor a felfutó/lefutó szakaszon a vezetés deriváltja lenne, ahol *G1* a magas hőmérséklethez tartozó vezetés, *G0* az alacsony hőmérséklethez tartozó vezetés. Ha a 6. ábrának megfelelő a vezetés görbéje, akkor ehhez még hozzá kell adni a felső és alsó görbe deriváltjának H-val arányos lineáris kombinációját.

Kódolás: a fázisváltó anyagok tulajdonság paraméterének value lekérdező függvénye legyen háromparaméteres:

Ge = anyag.Ge.value(T, H, m);

Ha az anyagparaméter nem broken line típusú, akkor a H és m értékét figyelmen kívül hagyja a value függvény. Lehetne egyértékű és kétértékű broken line típus, ahol a kétértékűt két egyértékűvel hozzuk létre.

Megjegyzés: *m* = 0, ha nem történik éppen fázisátalakulás. Azaz nem simán az 1/*TM*, hanem 0 vagy 1/*TM*.

**Jakobi mátrix**

Egy elektrotermikus cella modellje a 7. ábrán látható. Itt mindkét térre *n* db face csatlakozik, de ez akár különbözhet is. A referenciairány a normál face-eknél középpontba mutat, a középponti face-eknél pedig a föld felé.

Az elektromos és termikus földpont azonos, a termikus földet úgy állítjuk, hogy *Tamb* esetén 0 legyen. Az anyagparamétereket, peremfeltételeket stb. úgy kell létrehozni, hogy *Tamb*-bal inicializálhatók legyenek, és a szimuláció során ők a *Tamb*-tól való eltérést kapják hőmérsékletként.

A face-ek áramát a következő, általános egyenlet írja le:

(17)

Ahol *Ji* az *i*. ág (face) árama (elektromos/hő), beleértve a középponti face-eket is; *N* az ágak száma (itt 2*n*+2); *Vr* az ágfeszültség/hőmérséklet, azaz az ábrán *V*-vel és *W*-vel jelölt mennyiségek; *Gir* a konstans ellenállások vezetése (ha *i*=*r*) és a konstans transzfer vezetések (ha *i*≠*r*); *JGi*(*Vr*) az egy vagy akár több ágfeszültséggel/hőmérséklettel vezérelt nemlineáris áramforrás. Mivel az elektromos és a hőellenállás is hőmérsékletfüggő (bár sokszor konstansnak tekintjük), valójában a 7. ábrán az ellenállások helyett is vezérelt generátort lehetett volna rajzolni. Fontos megjegyezni, hogy bár a generátorokat ágfeszültségek ill. ághőmérséklet-különbségek vezérlik, a csomóponti potenciállal ill. a csomóponti hőmérséklettel is tudjuk vezérelni ezeket, ha létrehozunk egy ágat a kívánt csomópont és a föld között.

*Ge1*

*…*

*V1*

*I1*

*U1*

*Vn*

*In*

*Un*

*UC*

*VC*

*Gen*

*Gth1*

*…*

*W1*

*P1*

*T1*

*Wn*

*Pn*

*Tn*

*TC*

*WC*

*Gthn*

*Pd*

*7. ábra: Elektrotermikus cella modellje.*

A cella egyenlete (a Kirchhoff egyenletet alkalmazva (17)-re és *V*=*KU*-t helyettesítve a konstans tagba):

(18)

Ahol *K* az incidenciamátrix, *Gl* a konstans vezetés mátrix.

A Newton-Raphson módszer alkalmazásához az egyenlet bal oldalán a nullvektort a hibaáramok vektorára cseréljük:

(19)

A hibaáram az adott *U* ill. *V* esetén a csomópontba be és kifolyó áramok összege, ami Kirchhoff törvénye szerint 0 kellene legyen.

A Jakobi-mátrixot adja, és még felhasználva, hogy :

(20)

A *G* vezetésmátrix tehát a konstans vezetéseket, valamint az áramgenerátorok deriváltjait tartalmazza. Mivel az elektromos és hővezetések lehetnek hőmérsékletfüggők, a konstans vezetéseket el is hagyhatjuk:

(21)

Ha a vezetés mégis konstans, akkor is jó az egyenlet, hiszen akkor , ennek deriváltja szerint pedig .

A 7. ábrán bemutatott elektrotermikus cella vezetésmátrixa tehát (22) lesz. (Az 1 és *n* közötti sorokat és oszlopokat kihagyva.) Az elektromos ágban meghagytuk a vezérelt áramforrást, mert itt junction esetén ezt a formát érdemes használni (a junction árama nem arányos a feszültségével). Termikus esetben behelyettesítettük a hőellenállásokat: ezek árama arányos a rajtuk eső hőmérséklet-különbséggel.

(22)

A *VC* ág egy szakadás, így a sora nyilván csupa nulla, az oszlopa pedig azért csupa nulla, mert a cella potenciálja nem vezérel semmit (csak az ágfeszültségek). A *WC*-hez, azaz *TC*-hez tartozó oszlop viszont teljes, mert (a szakadást kivéve) minden áramköri elemnek lehet hőmérsékletfüggése. Ami konstans, annak ez az eleme 0 lesz. Az utolsó sorban azt látjuk, hogy a disszipáció az ágfeszültségektől függ, az ágak hőmérsékletkülönbségétől nem, de a cella hőmérsékletétől igen. (LED-eknél még egy disszipációs tényezővel szorzódhat a junction disszipációja, ami egy szorzó a 0…1 tartományban, de VO2-nél nincs ilyen). Amit lehet, kifejtve:

(23)

Az utolsó sorban azért negatívak a differenciális vezetések, mert a disszipált teljesítmény a cella középpontjába folyik be, a referenciairány viszont azzal ellentétes. (A referenciairány azért a középpontból a föld felé mutat, mert így kapjuk a középpont hőmérsékletét a környezeti hőmérséklethez képest.)

Az egyenletben szereplő *V* és *W* értékek az előző iterációban kapott teljes értékek, azaz nem az áramhibára kapott hibafesz/hőm!

Az incidenciamátrix:

(24)

A Jakobi-mátrix pedig, (21) alapján számítva:

(25)

, mert parciális deriváláskor a többi független változót konstansnak tekintjük.

Fontos különbség a klasszikus Sunred admittanciamátrixhoz képest, hogy ez nem szimmetrikus. Ha az ellenállások nem lennének hőmérsékletfüggők, a disszipáció miatt az aszimmetria megmaradna. Ekkor viszont nem lenne szükség Newton-Raphsonra, elég lenne a szukcesszív approximáció. A hibaáramok a (26) egyenlettel számolhatók:

(26)

A Jakobi-mátrix alapján a 8. ábrán látható differenciális cellamodell rajzolható fel.

*dUC*

*dVC*

*…*

*dV1*

*dI1*

*dU1*

*dVn*

*dIn*

*dUn*

*…*

*Gth1*

*dW1*

*dP1*

*dT1*

*Gthn*

*dWn*

*dPn*

*dTn*

*dWC*

*dTC*

*8. ábra: Elektrotermikus cella differenciális modellje.*

A 7. ábra modelljével összevetve az a különbség, hogy a 7. ábrán minden elem nemlineáris (is lehet), míg a 8. ábrán minden elem lineáris (konstans), de csak az aktuális iterációs lépésben érvényes, az előző lépésben kapott hőmérséklet esetén. Minden áramgenerátor modelljében két ágjellemző (feszültség ill. hőmérséklet-különbség) szorzata szerepel. A komponens azért lehet lineáris, mert ezek egyike az előző lépésben adódott érték, ebben a lépésben konstans. Az ebben a lépésben függő mennyiséget félkövér betűvel jelöltük. A termikus modellrész *TC* és föld között megjelent ellenállása azért ellenállásként szerepel, mert egy olyan termikus áramgenerátor, amit saját hőmérséklete vezérel, ami a termikus ellenállások jellemzője. (Azaz a rajta folyó hőáramot úgy kapjuk, ha az ellenállás mellé írt kifejezést *TC*-vel szorozzuk.) Ugyanezért szerepel ellenállás az elektromos modell face-eiben is.

A szukcesszív approximációhoz képest miben különbözik a Newton-Raphson módszerrel kapott modell?

Szukcesszív approximáció esetén a 7. ábrán szereplő modellt alkalmazzuk az adott lépésben, a lépés kezdetén érvényes hőmérsékletekhez és feszültségekhez tartozó ellenállásértékek és disszipáció mellett. A 8. ábra modelljének azért lesz jobb a konvergenciája, mert minden ellenállás mellett szerepel egy, az adott ellenállás hőmérsékletétől és nemlinearitásától függő áramgenerátor, ami az adott lépésben kiszámított feszültségek/hőmérsékletek figyelembevételével csökkenti ezek hibáját. A *TC* csomópontra kapcsolódó ellenállás pedig a disszipáció hibáját csökkenti. Kérdés, hogy ez elég lesz-e a VO2 erős nemlinearitásából adódó hiba akkora csökkentésében, ami konvergenssé tudja tenni a következő tranziens időlépés kiszámítását.

Fontos megjegyezni, hogy a Newton-Paphson számítás során, ellentétben a szukcesszív approximációval, maguk a feszültségek/hőmérsékletek/áramok is differenciális jellemzők. Tehát ha pl. *Tamb*=25°C, és a középpont 30°C, akkor a szukcesszív approximációban a *TC*=5°C körül ugrál a folyamatban, míg Newton-Raphson esetén csak az adott lépés első iterációjában lesz e körül, a következő iterációkban csak az előző iterációkban kapott érték és a legutolsó iterációban kapott érték különbsége lesz *TC*-ben.

Csak elektromos cella Jakobi-mátrixa:

(25b)

Csak termikus cella Jakobi-mátrixa:

(25c)

**A face modell**

Nem a kapcsolási rajzból, hanem a Jakobi-mátrixból indulunk ki. Minden face 9 kimeneti admittanciát produkálhat, melyek között lehetnek 0 értékűek:

**Junction modellek**

Az előző fejezetben azért nem került *GeiVi* a *Ji* helyére az elektromos face-ekben, mert tartalmazhat junctiont is, ami a saját ágfeszültségétől nemlineárisan függ. Ezt a viselkedést csak a junctionök számára engedjük meg. A junctiont tartalmazó face két részből áll (9. a) ábra): a tényleges junctionből és a hozzá vezető soros ellenállásból. Utóbbi nemlinearitását csak a hőmérsékletfüggés okozhatja. , , .

*Ge1*

*V1*

*I1*

*U1*

*VJ1*

*IJ1*

*Ub1*

*UC*

*Ve1*

*Ie1*

*TC*

*Pd*

*Vj1*

*I1*

*U1*

*Ve1*

*Ub1*

*UC*

*WC*

*TC*

*a)*

*b)*

*9. ábra: Junction modellek: normál és differenciális.*

A differenciális modellt a 9. b) ábra mutatja. Az *Ub1*-gyel jelölt belső csomópont kiejtése nemtriviális: rendes redukciós lépéssel érdemes csinálni. A termikus modellnél vegyük észre, hogy függőváltozóként is megjelenik az elektromos face belső *VJ1* és *Ve1* ágfeszültsége. Emiatt érdemes termikus junction face-eket is csinálni, és a disszipációt ott kezelni? Nem feltétlenül, de az alapcellák megvalósítását érdemes alaposan átgondolni.

*Új:*

1. Ágáramok számítsa előző ágfeszültségek alapján. Áram helyett a junction vezetését számítjuk ki az *U1-Ub1* feszültségből. A junction rész árama ennek feszültségszerese lenne, de ezt nem kell kiszámolni. A junction modell redukciójával is belátható, hogy az áram ill. hibaáram a teljes junctionra a (*GJ*×*Ge*)\*(*U1*-*UC*), azaz a két vezetés soros eredője szorozva a rajtuk eső feszültséggel adja a teljes junction áramát, amit innentől ugyanúgy kezelünk, mintha egy sima ellenállásunk lenne.
2. A teljes junction disszipációja a pn átmenet és az ellenállás disszipációjának összege. A pn átmenetnél azonban figyelembe kell venni a kisugárzott teljesítményt is. A teljes pn átmenet disszipációja így *Ij*(*Vj*,*TC*)\**Vj*\**D*(*TC*) - *F*(*Vj*,*TC*)\**R*(*TC*) + *Gr(TC)*\**Vr2*. *D* a disszipációs tényező, korrekcióra használható, általában 1 az értéke. *F* a sugárzott teljesítmény. *R* a sugárzási tényező, korrekcióra használható, általában 1.
3. Jakobi-mátrix: differenciális értékeket tartalmaz. A junction differenciális vezetése nem egyezik meg a sima vezetésével, ezért ezt a tulajdonság lekérdező függvénye külön adja. Nem tudom, használható-e, de (f×g)’ = (f2g’+f’g2)/(f+g)2.
4. A Jakobi-mátrix meghatározására a 9.a) ábra modelljét használjuk. Így az egyenlet:

Ábrán:

*Gr*

*V1*

*I1*

*U1*

*VJ*

*IJ*

*Ub*

*UC*

*Vr*

*Ir*

*TC*

*Pd*

*dV1*

*i1*

*u1*

*uC*

*dTC*

*tC*

*a)*

*b)*

*9. új ábra: Junction modellek: normál és differenciális.*

**Nemszimmetrikus hálózatredukció**

(27)

(28)

(29)

Kiszámolandó:

(30)

(31)

(32) \*XAT

(33)

(34) \*NZBXAT

(35)

(36) \*NZBXA

*XA* transzponált alakban kell.

*NZBXA* transzponáltját állítjuk elő az egyenlettel, ahol *NZB*-t nem kell transzponálni, mert úgy lesz sorfolytonos a szorzás. *YRED*-hez így lesz jó *NZBXA* formátuma a sorfolytonos szorzáshoz. *UB*-hez nem ideális, hogy *NZBXA* transzponált alakban van, de nem éri meg normál alakban is eltenni. Itt remélhetőleg nem vesztünk sokat, ha olyan formában végezzük a szorzást, hogy a transzponált mátrix első sorát szorozzuk a vektor első elemével, és ezt tesszük az eredményvektorba, majd a transzponált mátrix második sorát szorozzuk a vektor második elemével, és az így kapott szorzatokat hozzáadjuk az eredményvektor megfelelő elemeihez, stb. Azaz nem ugrálunk sem *NZBXAT*-ben, sem *UA*-ban, sem *UB*-ben. Sőt, *NZBJB* lehet a kiinduló vektor, és ehhez adogathatjuk hozzá -t.

**Impedancia modell**

Ha impedanciamátrixszal adottak a cellák, akkor a következőképpen alkalmazható a hálózatredukció.

(?)

Ahol:

* XA és XB ezúttal impedanciát tartalmaz.
* XA-ba és XB-be az első cellából pozitív, **a második cellából negatív előjellel** kell bemásolni az impedanciákat! (Admittanciamátrixnál mindkettő pozitív volt.)
* ZB a közös csomópontokhoz tartozó impedanciák összege, pont ahogy az admittanciánál.
* VB-be a két cella közös **inhomogén feszültségvektorainak különbsége** (V1-V2) kerül! (Admittanciánál az inh. áramok összege volt JB.)
* ZA és VA ugyanúgy másolódik, mint az admittanciánál a megfelelő mennyiségek.

Így jött ki:

* Felírjuk az U=ZI+V egyenletet mindkét cellára.
* Felbontjuk az egyenleteket [U1,U2]’=[A,B|C,D]\*[I1,I2]’+[V1,V2] ill. [U3,U4]’=[E,F|G,H]\*[I3,I4]’+[V3,V4] bontásban, ahol 2 és 3 a közös csomópontok, 1 és a szabadon maradók. Felhasználjuk, hogy I3 = -I2 és U3 = U2.
* Az admittanciához hasonlóan végigvisszük a levezetést.
* A levezetés szekennelve a mappában.

Ugyanazok az egyenletek, mint admittancia felírás esetén:

(28)

(29)

Tehát az előkészítés különbözik, a számítás ugyanaz.

**Centroid csomópontok**

A cellák modelljeiben megjelenhetnek „túl kicsi” és „túl nagy” ellenállások. Ez elsősorban az elektromos modelleket érintheti, de a termikus modelleknél is megjelenhet. Kialakulásuknak két oka lehet: fizikai és geometriai. A „túl kicsi” ellenállás fizikai oka, hogy az anyag nagyon jó vezető, így gyakorlatilag rövidzárnak tekinthető. A „túl nagy” ellenállás fizikai oka pedig nyilván az, ha az anyag jó szigetelő. Geometriai okból kicsi ellenállás akkor jön létre, ha a struktúra adott irányú keresztmetszetének kerülete és az adott irányú vastagságának aránya több nagyságrenddel különbözik (pl. 10 nm vastag réteg, 10 μm × 10 μm-es felület). A két hatást együtt érdemes kezelni, és a tényleges cellaellenállást figyelembe venni. Pl. egy vékony fémréteg laterális ellenállása könnyen lehet nem „túl kicsi” és nem „túl nagy”.

A „túl kicsi” és „túl nagy” ellenállásoknak kétféle negatív hatása van a számítások során. Egyrészt, ha az admittanciamátrixban (Jakobi-mátrixban) az elemek értékében 10-20 nagyságrend különbség van, különösen, ha ez a főátlót illeti, akkor a számítás numerikus pontossága romlik, akár kifejezetten hibás eredmény is adódhat. Másrészt feleslegesen számolunk, hiszen a „túl nagy” ellenállás gyakorlatilag szakadás, ha kihagyjuk a modellből, csökkentve az ágak számát, az nem változtatja érdemben a számítás eredményét. Ez úgy tudja csökkenteni a számítási időt, hogy a cellaközéppontokat összekötő ellenállásokat mindig két részre osztjuk, és a számítási idő lényegében a közöttük létrejövő csomópontok számának hatványával arányos. Ha elhagyunk egy ilyen ellenállást, akkor az osztóponti csomópontot is elhagyjuk, ezáltal csökkentve a számítási időt.

A „túl kicsi” ellenállás pedig rövidzár, azaz két vagy több szomszédos csomópontból egyet csinál, azaz szintén egyszerűsödik a hálózat.

Mit jelent mindez a sunred modell szempontjából? Mit a frame és mit a core szempontjából?

A „túl nagy” ellenállásokat a frame szintjén lehet kezelni, a core nem is látja ezeket. Adott cellában a „túl nagy” ellenállást elhagyjuk. Ha a csatlakozó face-ben megmaradna az ellenállás, azt is elhagyhatjuk. (Tulajdonképpen a frame elhagyhat minden open peremre csatlakozó ellenállást is, így egyszerűsítve a core dolgát, és a megmaradó ellenállás is egy ilyen open peremre csatlakozó face lenne.) Az ellenállások elhagyása és visszaállítása nemtriviális feladat a frame számára, de most a core-ra koncentrálunk.

A „túl kicsi” ellenállások nem kezelhetők csak a frame szintjén, mivel ha a helyükre simán rövidzárat tennénk a Jakobi-mátrixban, akkor az végtelen értékű admittanciát jelentene, ami nem kezelhető, meg nem is hoz előrelépést a sebességben sem, hiszen a csomópontok száma sem csökkenne.

Ha egy (vagy több) ellenállást rövidzárral helyettesítünk a cellamodellben, az a középső csomópont kivezetését jelenti, lásd a 10. ábrát. Ha egy cellán belül több ellenállást helyettesítünk rövidzárral (ez a tipikus, hiszen a szemben lévő ellenállások általában azonos értékűek), akkor ezeket az ellenállásokat egy darab kivezetett csomóponttal helyettesítjük, ezzel elkerülve a végtelen értékű admittanciát, ill. csökkentve az ágak, így a csomópontok számát. Az így kivezetett csomópontot centroid csomópontnak nevezzük. (Az elnevezést az analóg IC tervezésben használt „Common Centroid” tervezési módszer nevéből kölcsönöztem. A centroid jelentése egyébként súlypont, a cella középső csomópontja tipikusan tényleg a cella súlypontjára esik.)

*…*

*I1*

*U1*

*Vn*

*In*

*Un*

*UC*

*VC*

*Gen*

*…*

*P1*

*T1*

*Wn*

*Pn*

*Tn*

*TC*

*WC*

*Gthn*

*Pd*

*10. ábra: Centroid csomópontok. (Illusztráció, az elektromos és termikus centroid gyakran nem jár együtt!)*

Centroid csomópont esetén sem bonyolult a cella Jakobi-mátrixának előállítása: ugyanúgy előállítjuk a (25) egyenletet, csak nem végezzük el rá a hálózatredukciót, mivel nincs mit redukálni. Az így kapott, cellát mint zárt dobozt leíró Jakobi-mátrix annyiban különbözik a normál cella Jakobi-mátrixától, hogy maradnak benne 0 értékű elemek. Ez nem okoz gondot a további számítás során. (Az elektrotermikus cella Jakobi-mátrixában amúgy is szoktak maradni.) Ha egynél több ellenállást helyettesítünk rövidzárral egy cellán belül, akkor kevesebb lesz a külső csomópontok száma, azaz csökken a számítási idő. Egy helyettesítés esetén nem változik a számítási idő.

Ha egy elektrotermikus alapcella esetén pl. az elektromos térben létrejön centroid csomópont, a termikus térben nem, akkor a termikus középső csomópontot természetesen redukálni kell.

A centroid csomópont(ok) az admittanciamátrixon belül bárhol lehetnek, ahol a későbbi összevonás során indokolt. Nem kell ezeket az elejére/végére tenni, sem egymás mellé, ha több is van. (Természetesen lehet.) Ez a frame döntése, a core pedig tudja kezelni.

Abban az elfajult esetben, amikor egy cella összes (vagy összes elektromos/összes termikus) ellenállását rövidzárral helyettesítjük, a cella Jakobi-mátrixa egy darab admittancia, ami tipikusan 0 értékű (ha a középső csomópont és a föld között van valami, akkor nem). Ez csak akkor jelentene gondot numerikusan, ha lebegő részek lennének a térben, azaz szigetelővel körülvett vezető részek. Ezért 0 helyett érdemes egy kis értékű, (pl. 10-20 S) admittanciát alkalmazni. Ezt egyébként érdemes lehet minden cellaközépponton alkalmazni.

Mi lesz a centroid csomópontokkal, ha két cellát összevonunk?

Ezt az esetet mutatja a 11. ábra. Belátható (pl. az összevont cellák Jakobi-mátrixának felírásával), hogy az összevont Jakobi-mátrix ugyanúgy írható fel centroid csomópont összevonása esetén, mint nem centroid csomópont összevonása esetén, akár egyik, akár mindkettő, akár egyik sem centroid a két összevonandó csomópont közül. Egy különbség van: azok a csomópontok, amelyeket továbbra is ki kell vezetni (azaz centroidok maradnak), mert egy másik cellával is össze vannak kötve (több ellenállás lett rövidzárral helyettesítve az alapcellában), azokat nem redukáljuk. Ezeknek az admittanciáit csak összeadjuk ugyanúgy, mint normál redukciónál. Azaz algoritmikus szempontból csak a 12. ábrán látható eseteket kell megkülönböztetnünk.

*…*

*a)*

*…*

*b)*

*…*

*c)*

*11. ábra: Centroid csomópontok redukciója. a) Megmarad a centroid. b) Eltűnik a centroid. c) Összevonás normál cellával.*

A 12. b) ábra egy klasszikus cellaösszevonás redukcióval, algoritmikusan tehát teljesen mindegy, hogy a közös csomópontok bármelyike centroid-e vagy sem. Ha korábban a centroid csomópontot a Jakobi-mátrix megfelelő helyére tettük, akkor pont ugyanaz az algoritmus használható rá.

A 12. a) ábra esetén megmarad a centroid. Ez, az elemi cella Jakobi-mátrixának kiszámításához hasonlóan azt jelenti, hogy a közös csomópont admittancia értékei összeadódnak, de a csomópontot nem redukáljuk. Észrevehetjük, hogy ez az összevonás ugyanaz, mint amikor két cella egy-egy csomópontját rövidre akarjuk zárni ekvipotenciális felületként, pl. egy kompakt modellre kapcsoláshoz, vagy alternatív megoldásként, high-res regioként való használathoz. Az algoritmus valóban ugyanaz, tehát centroidként megjelölve a csomópontokat erre a célra is használható az algoritmusunk.

*a)*

*b)*

*12. ábra: Centroid összevonás cellaszinten. a) Megmarad a centroid = külső csomópontok rövidre zárása. b) Eltűnik a centroid = normál redukció.*

Az algoritmust elég úgy megcsinálni, hogy a két cellából egy-egy csomópontot von össze centroidként (ilyenből több is lehet, ahogy a 12. a) ábra mutatja), de egy-egynél többet nem, pl. a 12. a) ábrán a két fekete csomópontot már nem kell tudnia összekötni. Ez azért van így, mert ha a 12. a) ábrán lévő két centroid csomópontot még egymással is össze akarnánk kötni, akkor cellán belül az összevonásokat már korábban elvégezhettük volna, így ebben a lépésben már csak egy-egy csomópontot kéne összevonnunk, lásd 13. ábra. Ez számítási időben is hatékonyabb.

*a)*

*b)*

*13. ábra: Elég egy-egy csomópont centroid összevonását megvalósítani, így a) helyett b) módon oldjuk meg a problémát.*

**Blokkos invertálás**

Az első sor műveletei:

Ugyanezt a többi sorra is. A-E-I más-más méretűek is lehetnek, csak négyzetesek legyenek.

Ha az inverz –1-szeresét akarjuk megkapni, az aktuális oszlop (vagy sor) -1-szeresét számítjuk ki:

**Párhuzamos invertálás**

n = ((N+2)/4), m=N-3\*n.

A nemszimmetrikus inverz kiszámításának ütemezése:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | A.ninv(), A=>AT Q4-be | B=>BT Q1-be, C=>CT Q2-be, D=>DT Q3-ba, E=>R1-be, I=>R2-be, M=>R3-ba | | | | |
| 2: ha 1 kész | **a) E.nmul\_t (R1, Q4)** | b) I.nmul\_t (R2, Q4) | c) M.nmul\_t (R3, Q4) | d) B.nmul\_t (A, Q1) | d) C.nmul\_t (A, Q2) | d) D.nmul\_t (A, Q3) |
| 3: ha 2 a) kész | **a) F.sub\_mul\_t (F, E, Q1)** | | b) G.sub\_mul\_t (G, E, Q2) | | c) H.sub\_mul\_t (H, E, Q3) | |
| 4: ha 3 a) kész | **F.ninv()**, F=>FT Q4-be | | | | | |
| 5: ha 2 b) kész | a) J.sub\_mul\_t (J, I, Q1) | | b) K.sub\_mul\_t (K, I, Q2) | | c) L.sub\_mul\_t (L, I, Q3) | |
| 6: ha 2 c) kész | a) N.sub\_mul\_t (N, M, Q1) | | b) O.sub\_mul\_t (O, M, Q2) | | c) P.sub\_mul\_t (P, M, Q3) | |
| 7: ha 2, 3, 5 és 6 kész | E=>ET Q1-be, G=>GT Q2-be, H=>HT Q3-ba, B=>R1-be, J=>R2-be, N=>R3-ba | | | | | |
| 8: ha 4 és 7 kész | a) B.nmul\_t (R1, Q4) | **b) J.nmul\_t (R2, Q4)** | c) N.nmul\_t (R3, Q4) | d) E.nmul\_t (F, Q1) | d) G.nmul\_t (F, Q2) | d) H.nmul\_t (F, Q3) |
| 9: ha 8 b) kész | a) I.sub\_mul\_t (I, J, Q1) | | **b) K.sub\_mul\_t (K, J, Q2)** | | c) L.sub\_mul\_t (L, J, Q3) | |
| 10: ha 9 b) kész | **K.ninv()**, K=>KT Q4-be | | | | | |
| 11: ha 8 a) kész | a) A.sub\_mul\_t (A, B, Q1) | | b) C.sub\_mul\_t (C, B, Q2) | | c) D.sub\_mul\_t (D, B, Q3) | |
| 12: ha 8 c) kész | a) M.sub\_mul\_t (M, N, Q1) | | a) O.sub\_mul\_t (O, N, Q2) | | a) P.sub\_mul\_t (P, N, Q3) | |
| 13: ha 8, 9, 11 és 12 kész | I=>IT Q1-be, J=>JT Q2-be, L=>LT Q3-ba, C=>R1-be, G=>R2-be, O=>R3-ba | | | | | |
| 14: ha 10 és 13 kész | a) C.nmul\_t (R1, Q4) | b) G.nmul\_t (R2, Q4) | **c) O.nmul\_t (R3, Q4)** | d) I.nmul\_t (K, Q1) | d) J.nmul\_t (K, Q2) | d) L.nmul\_t (K, Q3) |
| 15: ha 14 c) kész | a) M.sub\_mul\_t (M, O, Q1) | | b) N.sub\_mul\_t (N, O, Q2) | | **c) P.sub\_mul\_t (P, O, Q3)** | |
| 16: ha 15 c) kész | **P.ninv()**, P=>PT S4-be | | | | | |
| 17: ha 14 a) kész | a) A.sub\_mul\_t (A, C, Q1) | | b) B.sub\_mul\_t (B, C, Q2) | | c) D.sub\_mul\_t (D, C, Q3) | |
| 18: ha 14 b) kész | a) E.sub\_mul\_t (E, G, Q1) | | b) F.sub\_mul\_t (F, G, Q2) | | c) H.sub\_mul\_t (H, G, Q3) | |
| 19: ha 14, 15, 17 és 18 kész | M=>MT S1-be,N=>NT S2-be, O=>OT S3-ba, D=>T1-be, H=>T2-be, L=>T3-ba | | | | | |
| 20: ha 16 és 19 kész | a) D.nmul\_t (T1, S4) | b) H.nmul\_t (T2, S4) | c) L.nmul\_t (T3, S4) | d) M.nmul\_t (P, S1) | d) N.nmul\_t (P, S2) | d) O.nmul\_t (P, S3) |
| 21: ha 20 a) kész | a) A.sub\_mul\_t (A, D, S1) | | b) B.sub\_mul\_t (B, D, S2) | | c) C.sub\_mul\_t (C, D, S3) | |
| 22: ha 20 b) kész | a) E.sub\_mul\_t (E, H, S1) | | b) F.sub\_mul\_t (F, H, S2) | | c) G.sub\_mul\_t (G, H, S3) | |
| 23: ha 20 c) kész | a)I.sub\_mul\_t (I, L, S1) | | b) J.sub\_mul\_t (J, L, S2) | | c) K.sub\_mul\_t (K, L, S3) | |

A szimmetrikus inverz kiszámításának ütemezése:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | A.sninv() | B=>BT Q1-be, C=>CT Q2-be, D=>DT Q3-ba | |
| 2: ha 1 kész | **a) E.nmul\_t (Q1, A)** | b) I.nmul\_t (Q2, A) | c) M.nmul\_t (Q3, A) |
| 3: ha 2 a) kész | **a) F.sub\_mul\_t (F, E, Q1)**, E=>ET B-be | b) G.sub\_mul\_t(G, E, Q2) | c) H.sub\_mul\_t(H, E, Q3) |
| 4: ha 3 a) kész | **F.ninv()** | | |
| 5: ha 2 b) kész | a) K.sub\_mul\_t (K, I, Q2) , I=>IT C-be | b) L.sub\_mul\_t (L, I, Q3) |  |
| 6: ha 2 c) kész | a) P.sub\_mul\_t (P, M, Q3) | b) M=>MT D-be | |
| 7: ha 3, 5 és 6 kész | B=> Q1-be, G=>GT Q2-be, H=>HT Q3-ba | | |
| 8: ha 4 és 7 kész | a) B.nmul\_t (Q1, F) | **b) J.nmul\_t (Q2, F)** | c) N.nmul\_t (Q3, F) |
| 9: ha 8 b) kész | **a) K.sub\_mul\_t (K, J, Q2)** | b) L.sub\_mul\_t (L, J, Q3), J=>JT G-be |  |
| 10: ha 9 b) kész | **K.ninv()** | | |
| 11: ha 8 a) kész | a) A.sub\_mul\_t (A, B, Q1) | b) C.sub\_mul\_t (C, B, Q2) | c) D.sub\_mul\_t (D, B, Q3) |
| 12: ha 8 c) kész | a) P.sub\_mul\_t (P, N, Q3) | b) N=>NT H-ba |  |
| 13: ha 9, 11 és 12 kész | C=>Q1-be, G=>Q2-be, L=>LT Q3-ba | | |
| 14: ha 10 és 13 kész | a) C.nmul\_t (Q1, K) | b) G.nmul\_t (Q2, K) | **c) O.nmul\_t (Q3, K)** |
| 15: ha 14 c) kész | **P.sub\_mul\_t (P, O, Q3)** | O=>OT L-be |  |
| 16: ha 15 kész | **P.ninv()** | | |
| 17: ha 14 a) kész | a) A.sub\_mul\_t (A, C, Q1) | b) B.sub\_mul\_t (B, C, Q2) | c) D.sub\_mul\_t (D, C, Q3) |
| 18: ha 14 b) kész |  | b) F.sub\_mul\_t (F, G, Q2) | c) H.sub\_mul\_t (H, G, Q3) |
| 19: ha 15, 17 és 18 kész | D=>S1-be, H=>S2-be, L=>S3-ba | | |
| 20: ha 16 és 19 kész | a) D.nmul\_t (S1, P) | b) H.nmul\_t (S2, P) | c) L.nmul\_t (S3, P) |
| 21: ha 20 a) kész | a) A.sub\_mul\_t (A, D, S1) | b) B.sub\_mul\_t (B, D, S2) | c) C.sub\_mul\_t (C, D, S3) |
| 22: ha 20 b) kész | a) F.sub\_mul\_t (F, H, S2) | b) G.sub\_mul\_t (G, H, S3) |  |
| 23: ha 20 c) kész | a) K.sub\_mul\_t (K, L, S3) |  |  |

**Fénypor modellezés**

A fény a Lambert-Beer törvény szerint nyelődik el egy anyagban:

Ahol a μ a hőmérsékletfüggő abszorpciós együttható, a d a réteg vastagsága.

A szimulációs modellnél három jelenséget veszünk figyelembe:

* A kék fény abszorpcióját a Lambert-Beer törvény szerint:   
   ,  
   .
* Az elnyelt\_kék-sárga konverziót:   
  ,  
   .
* A korábban keletkezett sárga fény abszorpcióját:   
   ,  
   ,  
   .

Ahol η a konverziós hatásfok.

Egy fénynyaláb a *C0* cella *F0* face-éből indul, áthalad *C1*, *C2*, …, *CN* cellán, míg megérkezik az aktuális *Cakt* cellába. Cakt cella adatai tehát a következők lesznek:

<*C0* száma, *F0* száma, a cellában keletkező (kék) fény mekkora részét veszi figyelembe ez az útvonal (*K0*)>.

<*C1* száma, *C1* figyelembe vett vastagsága (*d*), *C1* kék abszorpciós együtthatója, *C1* konverziós hatásfoka, *C1* sárga abszorpciós együtthatója, *C1* kimenő kékjének ezen az útvonalon figyelembe veendő aránya (*K1*, konstans), *C1* új sárgájának ezen az útvonalon figyelembe veendő aránya (*US1*, konstans), *C1* kimenő régi sárgájának ezen az útvonalon figyelembe veendő aránya (*RS1*, konstans)>. Ugyanez *C2*, …, *CN*-re.

< *Cakt* figyelembe vett vastagsága (*d*), *Cakt* kék abszorpciós együtthatója, *Cakt* konverziós hatásfoka, *Cakt* sárga abszorpciós együtthatója>

Az aktuális cella egy útvonalról érkező kék és sárga fénye, valamint disszipációja:

…

…

**Sárga útvonal**

Mivel egy modellben több millió kék sugár lehet, ha minden kék sugár minden sárga cellájához létrehoznánk több ezer sárga sugarat, akkor százmilliárd nagyságrendű sárga sugárral kellene dolgoznunk, ami nem megoldható.

Ezért lehetőség van arra, hogy a sárga sugarak a foszfor cellákból induljanak, és kiindulásként a foszfor cellákban keletkező összes új sárga sugarat használják.